Enhancing Human Action Recognition through Spatio-temporal Feature Learning and Semantic Rules

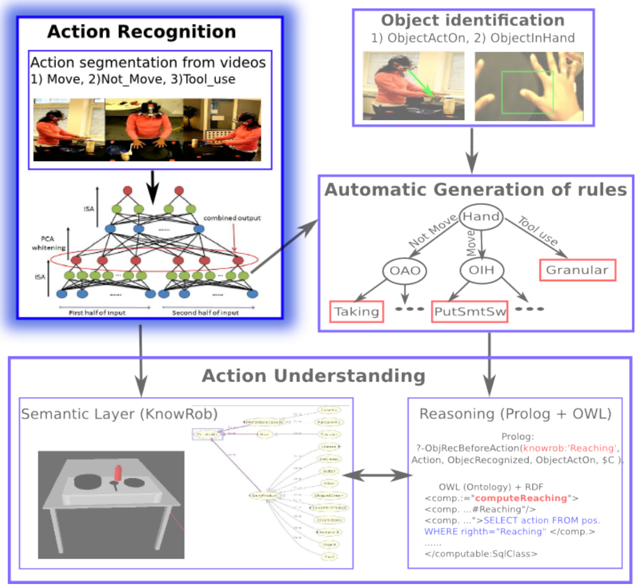
Karinne Ramirez-Amaro[1], Eun-Sol Kim[2], Jiseob Kim[2], Byoung-Tak Zhang[2], Michael Beetz[3]
and Gordon Cheng[1]

***Abstract*— In this paper, we present a two-stage framework that deal with the problem of automatically extract human activities from videos. First, for action recognition we employ an unsupervised state-of-the-art learning algorithm based on Independent Subspace Analysis (ISA). This learning algorithm extracts spatio-temporal features directly from video data and it is computationally more efficient and robust than other unsupervised methods. Nevertheless, when applying this one-stage state-of-the-art action recognition technique on the observations of human everyday activities, it can only reach an accuracy rate of approximately 25%. Hence, we propose to enhance this process with a second stage, which define a new method to automatically generate semantic rules that can reason about human activities. The obtained semantic rules enhance the human activity recognition by reducing the complexity of the perception system and they allow the possibility of domain change, which can great improve the synthesis of robot behaviors. The proposed method was evaluated under two complex and challenging scenarios: making a pancake and making a sandwich. The difficulty of these scenarios is that they contain finer and more complex activities than the well known data sets (*Hollywood2*, *KTH*, etc). The results show benefits of two stages method, the accuracy of action recognition was significantly improved compared to a single-stage method (above 87% compared to human expert). This indicates the improvement of the framework using the reasoning engine for the automatic extraction of human activities from observations, thus, providing a rich mechanism for transferring a wide range of human skills to humanoid robots.**

I. Introduction

One of the main purposes of humanoid robots is to improve the quality of live of elderly and/or disabled people by helping them in their everyday activities. Therefore, such robot systems should be flexible and adaptable to new situations. This means, they need to be equipped with capabilities of action recognition, action understanding, among others.

Regarding the problem of action recognition, the robot should be able to correctly identify through observations the actions and motions of the demonstrator. It is important to first identify the difference between the motions and activities that a human could do [1], this distinction will help to represent, recognize and learn human activities from videos. The state-of-the-art on human action recognition based on vision information identifies the local image representation as a promising direction compared to the global representations. The first one addressees visual occlusion and generalization to different scenarios by taking into account spatio-temporal correlation between patches [2]. There are four well known benchmark datasets: KTH, Hollywod2, UCF and YouTube and usually the action classes are: drive a car, eat, hand shake, hand wave, run, swing, etc, where the highest accuracy reported is around 75% [3]. Nevertheless, if we consider the actions that involve objects such as: basketball shot, golf swing, hand shake, answer a phone, etc, then we can notice that the accuracy of recognition is low (in average 43.62%). This shows how challenging those kind of activities are.



```
Prolog:
?-ObjRecBeforeAction(knowrob:'Reaching',
Action, ObjecRecognized, ObjectActOn, $C )

OWL (Ontology) + RDF
<comp.:="computeReaching">
<comp. ...#Reaching"/>
<comp. ...">SELECT action FROM pos.
WHERE righth="Reaching" </comp.>
......
</computable:SqlClass>
```

Fig. 1. Shows the overview of the approach proposed in this work. The highlighted section represents the data obtain from ISA and the dimmed section depicts the reasoning stage of the framework, similarly to [4].

Many directions for action recognition have been proposed, one of them is through the recognition of object(s) instead of observing human motions, e.g. Wörgötter et. al. [5] introduced the concept of Object-Action Complexes (OACs), which investigates the transformation of objects by actions, i.e. *how object A (cup-full) changes to object B (cup-empty) through the execution of Action C (drinking)*. This framework depends on the correct identification of the attributes of the objects. In a similar way, Patterson et. al. [6] presented a model that can be generalized from object instances to their classes by using abstract reasoning. Nevertheless, sometimes

[1] Faculty of Electrical Engineering, Institute for Cognitive Systems, Technical University of Munich, Germany karinne.ramirez@tum.de, gordon@tum.de. URL: www.ics.ei.tum.de

[2] School of Computer Science and Engineering, Seoul National University, South Korea {eskim, jkim, btzhang}@bi.snu.ac.kr

[3] Institute for Artificial Intelligence, University of Bremen, Germany beetz@cs.uni-bremen.de

the activities are misclassified because of the class of the object. Another direction is based on plan recognition [7] where is stated that human behavior follows stereotypical patterns that could be expressed as preconditions and effects. However, those constraints must be specified in advance.

A different approach is through human observations. In this work we follow this approach. Previous work presented by Gehrig et. al. [8] where their framework combines the motion, activity and intention recognition of a human by using visual information of a monocular camera in combination with the knowledge domain. This system is restricted to manual annotations of the domain (time of the day and the presence of the object). Also, the relationship between the activity and the motion is neglected. Furthermore, from human observations, it is also possible to extract the trajectory information and analyze its shape , with either linear-chain Conditional Random Fields (CRF) [9], or using Dynamic Time Warping [10]. Those classification techniques are restricted to the position of the objects in the trained environment, because if a different environment is analyzed then those trajectories will change completely and new models have to be acquired. Additionally, the object could be considered into the classification, where a library of Dynamic Motion Primitives (DMPs) enables the generalization of motions to new situations based on the new position of the goal [11], i.e. the motions are adjusted based on the parametrization of the given goal within the same neighborhood. This method takes perturbations into account and includes feedback [12].

In this paper, we present our approach to successfully recognize human actions from videos. We prove quantitatively the enhancement of the activity recognition using our reasoning engine. The key factor in our framework is the abstraction of the problem in two stages. First, by recognizing general motions such as *moving*, *not moving* or *tool used*. Second, by reasoning about more specific activities (*Reach*, *Take*, etc.) given the current context, i.e. using the identified motions and the objects of interest as input information. An illustration of the framework's pipeline is depicted in Fig. 1. In Section II, the Independent Subspace Analysis (ISA) will be introduced. In Section III, the methodology to create the rules is explained. In Section IV the data sets are described. Then, the results will be shown in Section V. Finally, Section VI will present the conclusions.

II. Motion Recognition of Human Demonstrators

In this work, we use the stacked Independent Subspace Analysis (ISA) algorithm to extract low-level features from videos, which will be later classified using Support Vector Machine (SVM) in order to recognize the human activities. The stacked ISA is an unsupervised learning technique that extracts invariant spatio-temporal features directly from unlabeled video data [3].

A. Stacked Independent Subspace Analysis (ISA)

Recent machine learning researches in the deep learning domain showed that the learning-based feature extraction algorithm is more effective than hand-designed visual feature algorithms such as SIFT, HOG, SURF [3]. The key advantage of those methods is that they are able to discover unexpected features. In contrast to the hand-designed features, which is totally based on the researcher's own heuristics.

The stacked ISA algorithm is a deep architecture consisting of several layers of ISA. It is often used to learn features from unlabeled image patches. The best way to describe this technique is as a two-layered network [13], where the first layer contains simple units with square non-linearities and the second layer is composed by pooling units with square-root non-linearities (see Fig. 2, a)).

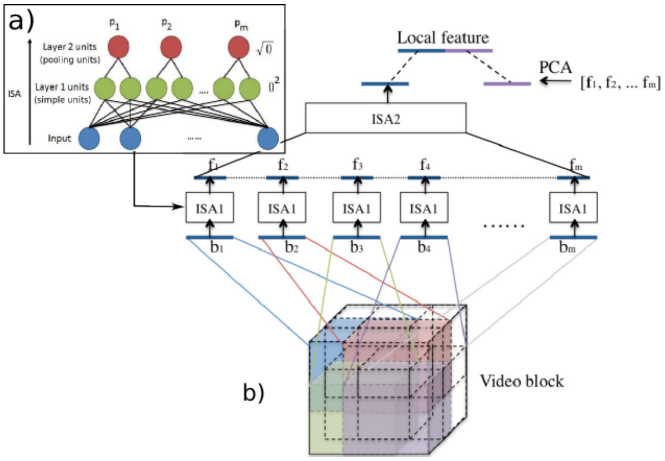


Fig. 2. a) shows the ISA neural network architecture and b) shows the stacked convolutional ISA for video data (Figure adapted from [3]).

The weights W in the first layer are learned, and the weights V of the second layer are fixed to represent the subspace structure of the neurons in the first layer. Each node of the second layer pools over a small neighborhood of adjacent first layer units. This means that given an input pattern x^t, the activation of each second layer unit is

$$p_i(x^t; W, V) = \sqrt{\sum_{k=1}^{m} V_{ik} (\sum_{j=1}^{n} W_{kj} x_j^t)^2} \tag{1}$$

where W is learned by finding sparse feature representations over the second layer with

$$\begin{aligned} \underset{w}{\text{minimize}} \; & \sum_{t=1}^{T} \sum_{i=1}^{m} p_i(x^t; W, V) \\ \text{subject to } & WW^T = I \end{aligned} \tag{2}$$

where $\{x^t\}_{t=1}^T$ are linearly transformed input examples and $W \in \Re^{k \times n}$ represents the weights connecting the input data to the simple units. $V \in \Re^{m \times k}$ defines the connection weights between the simple units and the pooling units. n, k and m are the input dimension, the number of simple units and the pooling units respectively. The orthonormal constraint is to ensure that the features are sparse enough.

This algorithm needs to be adapted for the video domain. The inputs to the network are 3D video blocks instead of image patches, i.e. we flatten the sequence of patches into a vector. This vector becomes the input features to a single ISA. Therefore, to learn high-level concepts it is necessary to stack several ISA networks. Then, a new convolutional neural network architecture is designed that progressively makes use of Principal Components Analysis (PCA) and ISA as subunits for unsupervised learning (see Fig. 2.b).

B. Human motion recognition methodology

We used a state-of-the-art processing pipeline similar to [3]. First, we learn spatio-temporal features using information from 3D video blocks as input. Those learned features are then convolved with a larger region of the input data. The outputs of this convolution are inputs to the next layer, which is also implemented by another ISA algorithm with PCA as prepossessing step to whiten the data and reduce its dimensionality. Then, the norm-thresholding is used to eliminate the features at locations where the activation norm is below the defined threshold (δ), i.e. this threshold will filter out features from the non-informative background. In this work, we choose $\delta = 30\%$. Finally, in our experiments we combine the extracted interesting features from both layers and use them as local features for classification using a χ^2-kernel Support Vector Machine (SVM). This methodology is summarized in the video that is attached to this paper.

As a result of the above methodology, three human motions will be recognized: 1) Move, 2) Not Move and 3) Tool Use[1]. A set of the recognized hand human motions, produces an *Activity* and a set of activities, such as Reaching, Taking, Releasing, etc., defines a *Task*, e.g. cut the bread.

1) Experimental ISA set-up: In order to learn the spatio-temporal features, we use images from random video blocks of size 16×16 (spatial[2]) and 10 (temporal[3]). Additionally, we set the input dimension and the number of simple units as $k = m = 300$. This means that the input of our first ISA layer learns 300 features. Then, the inputs to the second layer are defined of size 20×20 (spatial) and 14 (temporal). The simple units of the stacked ISA are set to $k = 200$ and the pooling units are set to $m = 100$, i.e. the second layer ISA network learns 200 features.

III. Automatic Generation of Rules

The segmented motions obtained from the above procedure represent the abstract level of our model. Our goal is to identify the human basic activities, and we propose a novel methodology to handle this problem by inferring the activities based on the observed human motions and the information of the object of interest (environment information). Therefore, the information of the objects involved in the activity is also considered and they could be[4]:

1) Object Acted On[5], 2) Object In Hand[6].

Finally, from the above information, the robot should be able to infer the activity that the human is performing by automatically obtaining those inference rules. In this work, we proposed to obtain those rules with a decision tree based on the C4.5 algorithm [14]. This algorithm is very robust to noisy data and it is able to learn disjunctive expressions. Decision trees provides a very reliable technique to learn top-down inductive inference rules (*if-then* rules). The central core of the C4.5 algorithm is to select the most useful attribute to classify as many examples as possible by using the information gain measure, defined as:

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{S} Entropy(S_v) \quad (3)$$

where $Values(A)$ is the set of all possible values of the attribute A, and $S_v = s \in S | A(s) = v$ for a collection of examples S. The entropy is defined as:

$$Entropy(S) = \sum_{i=1}^{c} -p_i log_2 p_i \quad (4)$$

where p_i is the probability of S to belong to class i.

IV. Task Examples

In order to test the robustness of the generated rules in different scenarios, we decided to use two real-world scenarios, i.e. *making a pancake* and *making a sandwich*. These activities present different levels of complexity and they involve different kinds of objects as we will show in the next sub-sections.

A. Making pancakes

In our first scenario, we recorded new videos of humans performing the activity of making a pancake. We choose, this realistic cooking activity, because it allow us to analyze goal-directed movements, which contain finer humans motions than the typical data sets. We recorded one participant performing the action nine times. These recordings contain information of three external cameras at 24 fps and the cameras are located in different positions (see Fig. 3).

B. Making a sandwich

The second experimental scenario contains a more complex activity, which is making a sandwich. These recordings also contain the information of three external cameras at 60 fps. Additionally, this task is performed by two subjects and each subject prepared two sandwiches, one sandwich was prepared under normal time condition and the second one under time pressure (in a hurry). One can see from Fig. 4 that this activity contains several objects as well as different tasks. This means that this scenario is more complex than the pancake making scenario. It is important to notice that some activities are performed simultaneously from both the right hand and the left hand, for example, Fig. 4.2) shows that the left hand is holding the bread while the right hand is cutting with a knife.

[1] This complex motion involves two objects, one is used as a tool and the other is the recipient of the action, e.g. *pouring* or *cutting*.

[2] Spatial refers to the pixel dimensions of the image patches.

[3] Temporal represents the frames per second used to define a video.

[4] As a first approach, we will use the object information from manually annotated videos.

[5] Identifies the object which will be manipulated.

[6] Defines the object that is physically in the hand, i.e. the object that is currently manipulated.

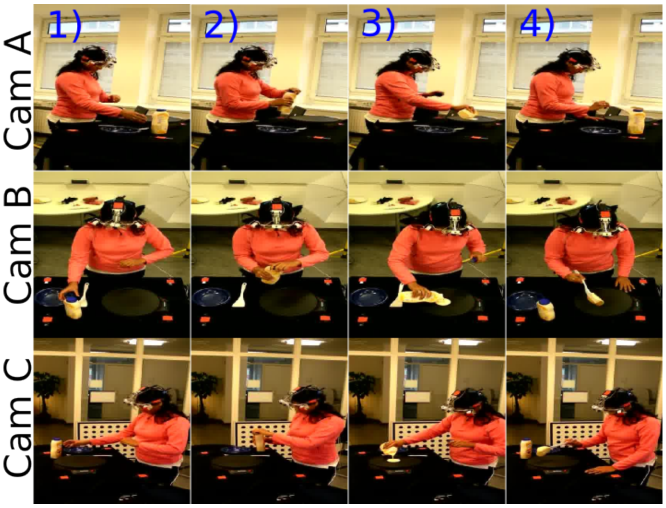


Fig. 3. This figure shows the three external cameras and some examples of the main activities involved in making a pancake performed by the right hand: 1) Reach, 2) Hold, 3) Pour, 4) Flip.

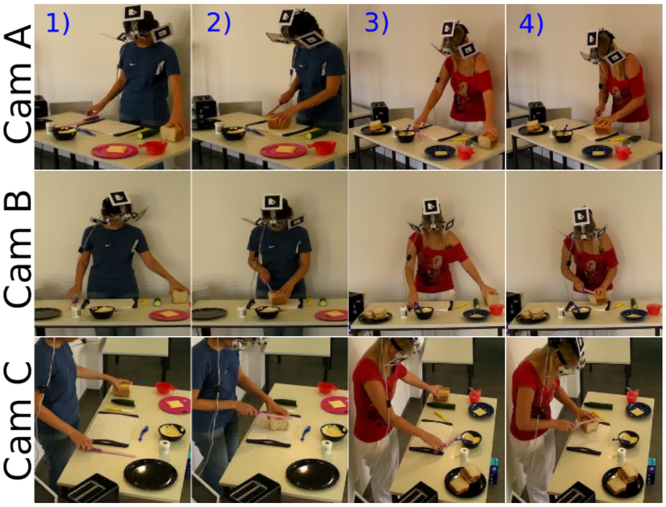


Fig. 4. This figure depicts the sandwich making scenario made by two subjects. A subset of activities executed with the right hand are shown: 1) and 2) present the activities performed by Subject 1: Reach and Cut. 3) and 4) show similar activities performed by subject 2.

V. Results

This section presents the obtained results and it is divided into two subsections. The first one presents the results from the human motions recognition from videos using the extended ISA algorithm. The second part shows the rules extracted from the human observations and their robustness in different data sets.

A. Action recognition

First, we present the results from the pancake making video dataset (see Fig. 3). During the training phase, we used the information of the three-view cameras for the first subject during the first trial. Afterward, for testing, we used another video from a different trial. We would like to stress that people can not perform the same activity identically. Hence, we can not expect a 100% accuracy in the activity recognition. The results present that the correctly recognized human high-level motions (such as move, not move and tool use) is 81%, which represents a very high accuracy, even when the videos were very short because the cameras were at 24 fps. The obtained confusion matrix[7] can be observed in Table I a).

Then, we tested the sandwich making video dataset (see Fig. 4). This video contains more objects in the scene and more motions are used during the preparation of a sandwich. Additionally, in this particular set-up, we constrained the speed of the preparation of the sandwich. This means that participants perform the first sandwich in normal speed and the second in high speed (simulating they were in a hurry). This means the variance between the trials is high, even when they are performed by the same subject. Therefore, we use for training the three-view videos from the first participant during the normal condition and, for the testing stage, we use the same participant but with the high speed condition. Hence, the correctly classified human motions is 71%, which is also very high, compared with the single-stage method [3]. Please refer to Table I b) to see the obtained confusion matrix. Additionally, it could be observed that *tool used* motions are misclassified as *move* motions, because *tool use* could be considered as a subclass of *move*, therefore we may need the information of the objects to help the system to distinguish between this two different motions.

TABLE I

Confusion matrix results for pancake and sandwich making

	a) Pancake making			b)Sandwich making		
Classified as Actual Class	Move	Not Move	Tool Use	Move	Not Move	Tool Use
Move	**6**	6	0	**13**	2	**6**
Not Move	0	**10**	0	1	**9**	1
Tool Use	0	1	**2**	0	0	5

Afterward, we used the obtained features from the first subject and we tested them on the second subject and the classification accuracy for the high-level human motions is around 65%, which is high considering that we used different video sets for training and testing. This means that the video features between the videos are different, for example, the subjects have different shirt color and that feature has not been previously trained. This represents a very important aspect which is considered as a *self-taught learning* problem. To the best of our knowledge the state-of-the-art techniques for action recognition can achieve about 51.5% of accuracy for the *self-taught learning* [3]. It is important to mention that if, instead of classifying the high-level motion, we would have classified directly the low-level activities (e.g. Reach, Take, Release, etc.), then the classification performance would have decreased abruptly to below 25%, considering the same video sets for training and testing as before. Classifying the low-level activities is an example of the single-stage method [3].

[7] This confusion matrix is computed for each sub-video, i.e. for the pancake testing scenario we have 25 sub-videos of different length.

Another important advantage of using this model is the reduced time required during the training process to 1-2 hours[8], when the normal required time is 2-3 days. This is because the stacked convolved network model is trained greedily layer-wise in the same manner as other algorithms proposed in the deep literature [15]. This means that the layer 1 was trained until convergence before training layer 2.

B. Automatic generation of rules

First, we need to build a decision tree as general as possible in order to capture the relationships between the objects, motions and activities to correctly infer the human basic activities under different scenarios. Since this tree is computed only once, we use as input information the labels obtained from the manual annotation of the pancake making videos executed by one subject during the first trial. The weka data mining software was used to generate the decision tree [16]. We only used the first 30% of the whole pancake making video to build the tree and the rest (70%) was used for testing. The tree that we obtained from the training data set is shown in Fig. 5 and the testing data set was correctly classified in 95.87%.

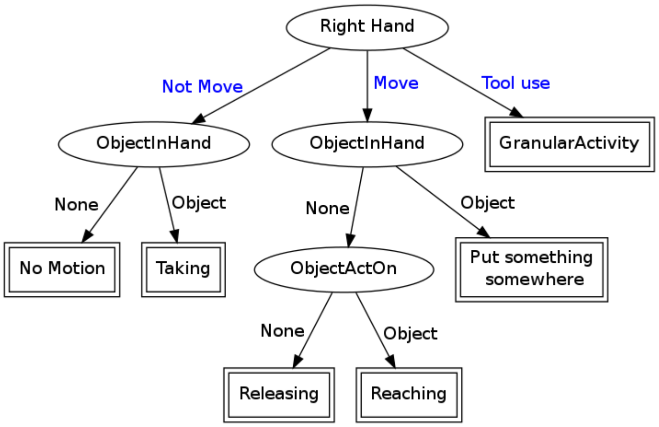


Fig. 5. This figure shows the obtained decision tree. Notice, that a correct activity classification depends on the accurate recognition of the human motion (blue letters).

From the above tree, it is possible to obtain some of the following rules:

$$if\ \ RightHand(Not\_Move)\ and\ ObjectInHand(Object) \rightarrow Activity(\textbf{Taking}) \tag{5}$$

$$if\ RightHand(Move)\ and\ ObjectInHand(None)\ and\ ObjectActedOn(Object) \rightarrow Activity(\textbf{Reaching}) \tag{6}$$

$$if\ RightHand(Move)\ and\ ObjectInHand(Object) \rightarrow Activity(\textbf{PuttingSomethingSomewhere}) \tag{7}$$

The pancake making action has principally three main tasks[9]: *Pouring*, *Flipping* and *Sliding out*. From the tree, one can observe that these sub-activities are clustered using the same rule:

$$if\ RightHand(Tool\_use) \rightarrow Activity(\textbf{GranularActivity}) \tag{8}$$

These activities will not be considered as human *basic* activities, therefore in order to correctly classify those activities more information is needed. In the remainder of this paper this kind of activities will be called *granular activities*. One possibility to classify those *granular activities* is to use all the activities classified as *Tool use* and sub-classify them again using the ISA methodology explained in Section II.

The next step is to test the robustness of the obtained tree (see Fig. 5) by using as input the classification results obtain from the ISA algorithm from the videos. Therefore, first we used the motion recognition results obtained from the extended ISA algorithm for the pancake making videos. The correctly classified instances are 93.60% and the final human activity recognition after the two stages is around 87.3%. Second, we used as new input the classification results for the sandwich making videos. Please take into account that the tree was generated from a 24 fps video and the new test set contains information at 60 fps. In this case, we used the motion recognition results obtained from the testing data set, which corresponds to subject 2 with the high speed condition (the normal speed condition was used as training for the ISA algorithm). Please notice that the recognition results from ISA are lower than in the pancake making data set. Hence, the correctly classified instances using the tree from Fig. 5 are 81.1%, this is affected by the errors produced by the incorrect classification generated by the ISA algorithm. Therefore, the final action recognition is 76.05%. The confusion matrix[10] can be observed in Table II.

TABLE II
CONFUSION MATRIX FROM THE ACTIVITY OF SANDWICH MAKING

Actual Class \ Classified as	a	b	c	d	e	f
a)Idle motion	**3**	0	0	0	2	3
b)Reach	20	**190**	1	8	12	24
c)Take	0	4	**56**	12	0	28
d)PutSomethingSomewhere	0	0	47	**450**	1	224
e)Release	0	1	1	3	**72**	27
f)Granular	0	0	0	21	0	**1113**

Afterward, we used the motions recognized as *granular activities* (see Table II.f) and re-classify them following the ISA methodology. In the case of the sandwich making data set, the *granular activities* are one of the following categories: cutting, unwrapping, spreading and sprinkling and the classification accuracy is 72.2%.

Using this methodology the complexity of action recognition decreases, because we propose to first classify the high-level activities and then use that information to infer the

[8] The experiments were carried out with a PC-desktop with 8GB RAM and Intel® Core$^{\text{TM}}$ i7.

[9] A task (sub-activity) is defined as a motion where a tool is used, such as flipping or cutting something, where the tools are the spatula or the knife, respectively.

[10] This confusion matrix is obtained frame-wised.

low-level activities. If we want to classify this low-level activities, the classification accuracy will decrease substantially, because certain activities like Reaching will be misclassified as Releasing due to their similarity.

The rules obtained from the decision tree can be programed easily in any kind of language. Nevertheless, some first order logic languages such as Prolog, could enhance the system with more inference and reasoning capabilities. By reasoning we mean that some facts will be derived (infer), these facts are not necessarily expressed in the ontology or in the knowledge base explicitly. Therefore, as part of our framework, those rules represent an important part of our reasoning engine, because they will help to recognize human everyday activities. Hence, we introduce new features to our reasoning engine to infer the new relationships between motions, objects and activities. The description of our ontology-based reasoning engine can be found in [4].

C. Results summary

An important aspect of our work that needs to be highlighted is that, on one hand, if the human action recognition for complex and real-world scenarios like the ones presented here is classified using the classical approach, i.e. trying to recognize the low-level activities, such as: reaching, taking, releasing, cutting, sprinkle, etc., then the classification accuracy will be around 25% (see Section V-A). This is the classical approach that is used for action recognition, where the state-of-the-art techniques for goal-directed activities reported an accuracy of about 43.6% [3].

On the other hand, if the method introduced here is used, i.e. split the classification problem into first recognizing the three high-level motions (move, not move and tool use), and later use the classification output as input into the reasoning engine, then the action recognition will be inferred, increasing the accuracy to approximately 82%. Therefore, with our proposed framework the activity recognition accuracy increases by 55% than with the classical approach using the same datasets. Even though the results presented here are for off-line classification, the extension to on-line classification is possible. This is because once the video features are extracted from the training process, the classification results are obtained in seconds. This extension is considered as future work.

VI. Conclusions

The correct identification of human activities is a challenging task in the robotics community and its solution is very important because it is the first step toward a more natural human-robot interaction. In this paper we presented a methodology to correctly recognize human activities from videos, by splitting the complexity of the activity recognition problem into first classifying the high level human activities from videos using the extended ISA algorithm and afterward inferring the human activities with the semantic rules. Those rules have the important characteristic that they can be used in different scenarios. This represents our first approach to find rules that could generalize human basic activities.

If the robot is able to correctly identify the activities from his/her demonstrator, then it will probably be able to predict his/her next motion or intention. In this paper, we present our first approach to tackle the first part of that problem and we prove that with this methodology different scenarios can be considered. The second part is considered as future work.

Acknowledgments

K. Ramírez-Amaro is supported by a CONACYT-DAAD scholarship and GEnKO. Eun-Sol Kim and Jiseob Kim are supported by the National Research Foundation (NRF-2012-0005643-Videome, NRF-2011-0020997-GEnKO).

References

[1] P. K. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine Recognition of Human Activities: A Survey." *IEEE Trans. Circuits Syst. Video Techn.*, vol. 18, no. 11, pp. 1473–1488, 2008.

[2] R. Poppe, "A survey on vision-based human action recognition." *Image Vision Comput.*, vol. 28, no. 6, pp. 976–990, 2010.

[3] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis." in *CVPR*. IEEE, 2011, pp. 3361–3368.

[4] K. Ramirez-Amaro, M. Beetz, and G. Cheng, "Extracting Semantic Rules from Human Observations." in *ICRA'13 workshop: Semantics, Identification and Control of Robot-Human-Environment Interaction. 2013 IEEE International Conference on Robotics and Automation.*, May 2013.

[5] F. Wörgötter, A. Agostini, N. Krüger, N. Shylo, and B. Porr, "Cognitive agents - a procedural perspective relying on the predictability of Object-Action-Complexes (OACs)." *Robotics and Autonomous Systems*, vol. 57, no. 4, pp. 420–432, 2009.

[6] D. J. Patterson, D. Fox, H. A. Kautz, and M. Philipose, "Fine-Grained Activity Recognition by Aggregating Abstract Object Usage." in *ISWC*. IEEE Computer Society, 2005, pp. 44–51.

[7] H. A. Kautz, H. A. Kautz, R. N. Pelavin, J. D. Tenenberg, and M. Kaufmann, "A formal theory of plan recognition and its implementation," in *Reasoning about Plans*. Morgan Kaufmann, 1991, pp. 69–125.

[8] D. Gehrig, P. Krauthausen, L. Rybok, H. Kuehne, U. D. Hanebeck, T. Schultz, and R. Stiefelhagen, "Combined intention, activity, and motion recognition for a humanoid household robot." in *IROS*. IEEE, 2011, pp. 4819–4825.

[9] M. Beetz, M. Tenorth, D. Jain, and J. Bandouch, "Towards Automated Models of Activities of Daily Life," *Technology and Disability*, vol. 22, 2010.

[10] S. Albrecht, K. Ramirez-Amaro, F. Ruiz-Ugalde, D. Weikersdorfer, M. Leibold, M. Ulbrich, and M. Beetz, "Imitating human reaching motions using physically inspired optimization principles." in *Humanoids*. IEEE, 2011, pp. 602–607.

[11] A. Ude, A. Gams, T. Asfour, and J. Morimoto, "Task-Specific Generalization of Discrete and Periodic Dynamic Movement Primitives." *IEEE Transactions on Robotics*, vol. 26, no. 5, pp. 800–815, 2010.

[12] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Movement Imitation with Nonlinear Dynamical Systems in Humanoid Robots." in *ICRA*. IEEE, 2002, pp. 1398–1403.

[13] A. Hyvärinen, J. Hurri, and P. O. Hoyer, "Natural Image Statistics: A Probabilistic Approach to Early Computational Vision," 2009.

[14] R. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann Publishers, 1993.

[15] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations." in *ICML*, ser. ACM International Conference Proceeding Series, A. P. Danyluk, L. Bottou, and M. L. Littman, Eds., vol. 382. ACM, 2009, p. 77.

[16] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009.